# Probleemoplossing en debug VoIP-oproepen

## Inhoud



## Inleiding

In dit document worden de basistechnieken en -opdrachten beschreven om VoIP-netwerken op te sporen en te zuiveren.

## Voorwaarden

### Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- VoIP-configuratie

- Spraak-QoS
- Ontwerp en implementatie van VoIP-netwerken

### Gebruikte componenten

Dit document is niet beperkt tot specifieke software- en hardware-versies. De getoonde uitgangen zijn echter gebaseerd op Cisco IOS®-softwarerelease 12.3(8).

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

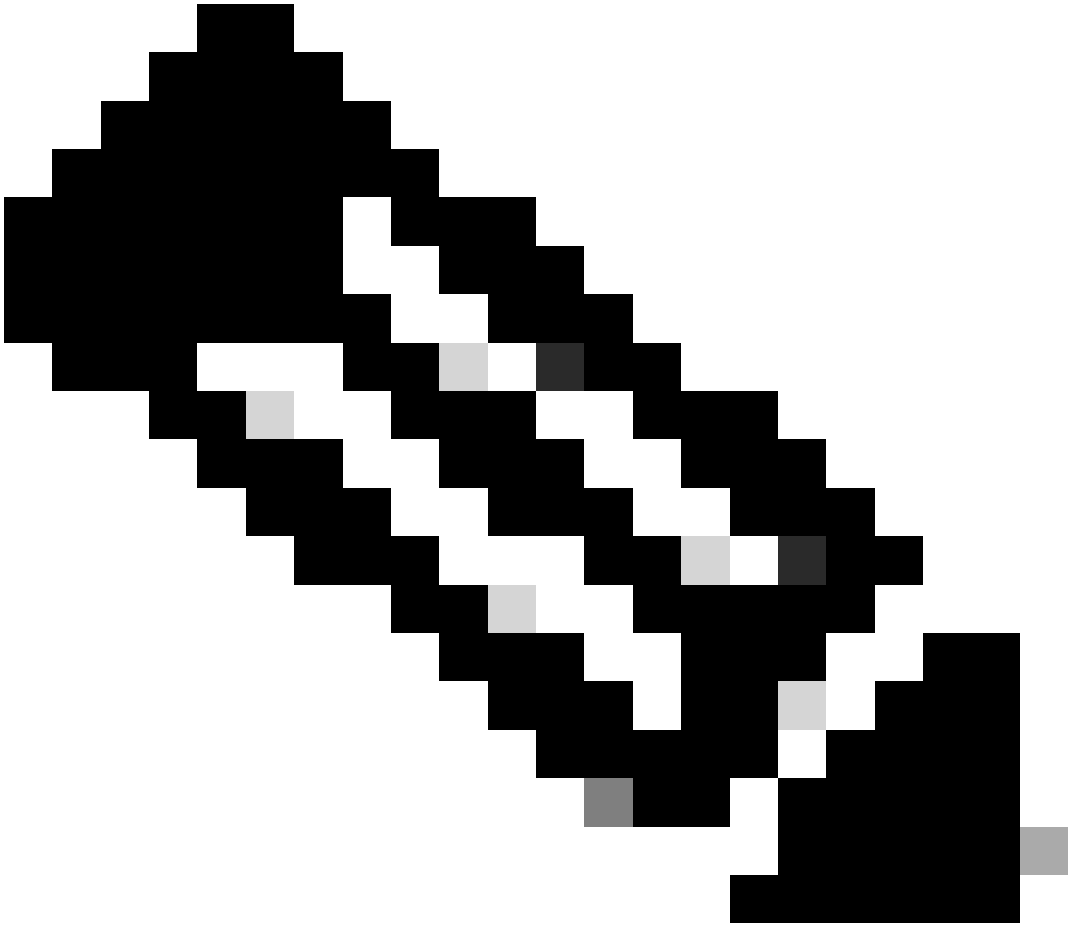
### Conventies

Raadpleeg Cisco Technical Tips Conventions (Conventies voor technische tips van Cisco) voor meer informatie over documentconventies.

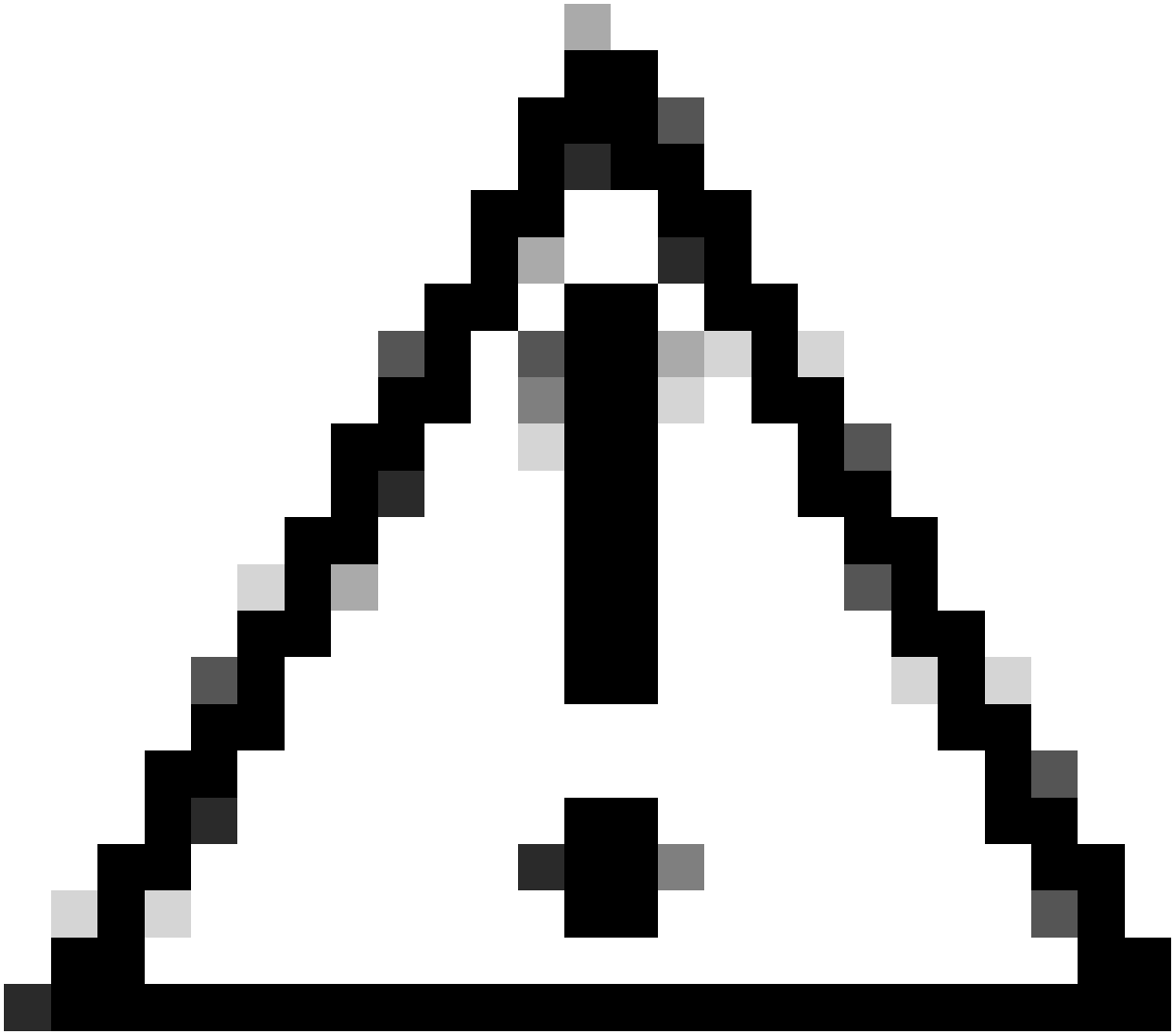
## Achtergrondinformatie

In dit document worden basistechnieken en -opdrachten beschreven voor troubleshooting en foutopsporing van VoIP-netwerken. Er is een overzicht van de architectuur voor gespreksstromen en telefonie in een Cisco-router opgenomen, alsmede een stapsgewijze aanpak om VoIP-problemen te troubleshooten:

1. Controleer digitale en analoge signalering.
2. Controleer de cijfers die van de analoge en digitale spraakpoorten worden ontvangen en verzonden.
3. Controleer end-to-end VoIP-signalering.
4. Begrijp VoIP Quality of Service (QoS)-problemen.
5. Begrijp details van oorzaakcodes en debug waarden voor VoIP.



Opmerking: dit document verklaart niet elk facet van de Cisco IOS-architectuur die wordt gebruikt in Cisco VoIP-gateways en gatekeepers. In plaats daarvan is het bedoeld om te laten zien welke opdrachten kunnen worden gebruikt en welke velden uit de opdrachtoutput het meest waardevol kunnen zijn.



Waarschuwing: het is processorintensief om Cisco IOS te debuggen. Voorzichtigheid is geboden wanneer u de debugs gebruikt die in dit document worden weergegeven. Raadpleeg Belangrijke informatie over debug-opdrachten voor meer informatie.

Debugs moeten worden uitgevoerd met timestamps ingeschakeld in het logbestand. Schakel timestamping met de opdrachten in: service-timestamps debug datetime msec, service-timestampslog datetime msec in inschakelen modus. De tijdstempels helpen bij het bepalen van het tijdsinterval tussen de wijzigingen in de status.

## Call Flow in het netwerk

Een belangrijke factor die u in overweging moet nemen voordat u met VoIP-probleemoplossing of debugging begint, is dat VoIP-oproepen bestaan uit drie gespreksbenen. Deze telefoonbenen zijn POTS-systemen (Source Plain Old Telephone Systems), VoIP en POTS-doelsystemen. Dit is in dit diagram te zien. Problemen oplossen en debuggen moet zich eerst onafhankelijk op elk been richten en vervolgens op de VoIP-oproep als geheel.

## Routergespreksstroom

Deze definities verklaren de functie van de belangrijkste componenten die in het stroomschema van de routervraag worden getoond:

Call Control API (Application Programming Interface)—Drie clients maken gebruik van de Call Control API. De drie clients zijn CLI, Simple Network Management Protocol (SNMP)-agent en de Session Application. De hoofdfuncties van Call Control API (ook wel CCAPI genoemd) zijn:

- Identificeer de vraagbenen (bijvoorbeeld, welke wijzerplaat peer is het? Waar kwam het uit?).
- Beslis welke sessietoepassing de oproep doet (bijvoorbeeld wie behandelt deze?).
- Pak de pakkethandler aan.
- Confereer de aanroepbenen samen.
- Begin om vraagstatistieken te registreren.

Session Application en Dial Plan Mapper—De Session Application gebruikt de Dial Plan Mapper om een nummer toe te wijzen aan een dial-peer (lokale POTS of externe VoIP). De kiesschema-map gebruikt de tabel met dial-peers om actieve dial-peers te vinden.

Telefonie en VoIP Service Provider Interface (SPI)—De SPI voor telefonie communiceert met de POTS-peers (analoog: fxs, fxo, e&m Digital: ISDN, qsig, e&m, enzovoort). De VoIP SPI is de specifieke interface voor de VoIP-peers. Stuurprogramma's voor telefonie/DSP leveren services aan de SPI voor telefonie terwijl VoIP SPI afhankelijk is van sessieprotocollen.

## Telefonie-interfacearchitectuur

Dit diagram toont de architectuur van de bouwstenen van Cisco-routertelefonie en de manier waarop ze met elkaar communiceren.

In deze lijst worden de functies en definities van de hoofdcomponenten van het diagram beschreven:

- Call Control Application Programming Interface (CCAPI)—Software-entiteit die benen maakt, beëindigt en overbrugt.
- Voice Telephony Service Provider (VTSP)—Een Cisco IOS-proces dat serviceaanvragen ontvangt van de Call Control API en de juiste aanvragen formuleert naar de digitale signaalprocessor (DSP) of de VPM.
- Voice Processor Module (VPM)—De VPM is verantwoordelijk voor het overbruggen en coördineren van signaleringsprocessen tussen de telefoniepoorten en de signaleringsstaatsmachine (SSM), de DSP Resource Manager en de VTSP.

- DSP Resource Manager—De DSPRM biedt interfaces waarmee de VTSP berichten kan verzenden en ontvangen naar en van de DSP's.
- Packet Handler - De pakkethandler stuurt pakketten door tussen de DSP's en de peer Call-benen.
- Call Peer - de Call Peer is het tegenovergestelde gespreksbeen. Dit kan een andere telefoonspraakverbinding (POTS), VoFR, VoATM of een VoIP-verbinding zijn.

# Verifieer digitale en analoge signalering (POTS-gespreksleuf)

Het verifiëren van digitale en analoge signalering is:

- Bepaal dat de juiste analoge of digitale on-haak en off-haak signalering is ontvangen.
- Bepaal dat juiste E&M-, FXO- en FXS-signalering is geconfigureerd op zowel de switch als de router (CO of PBX).
- Controleer of de DSP's in de verzamelmodus voor cijfers staan.

De opdrachten die in deze secties worden beschreven, kunnen worden gebruikt om de signalering te controleren.

## toon controllers T1 / E1 (digitaal)

toon controllers t1 [sleuf/poort]—Gebruik deze opdracht eerst. Het toont als de digitale T1 verbinding tussen de router en de switch (CO of PBX) omhoog of omlaag is en als het behoorlijk functioneert. De uitvoer van deze opdracht ziet er als volgt uit:

```
<#root>

router#

show controllers T1 1/0

T1 1/0 is up.

Applique type is Channelized T1
Cablelength is short 133

No alarms detected.
Framing is ESF, Line Code is B8ZS, Clock Source is Line
Primary.

Data in current interval (6 seconds elapsed):

    0 Line Code Violations, 0 Path Code Violations
    0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs,
  0 Degraded Mins
    0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs,
  0 Unavail Secs
```

Als u E1 gebruikt gebruik de show controllers e1 opdracht. Zie voor meer informatie:

- T1 Layer 1-probleemoplossing
- T1-stroomschema voor probleemoplossing
- Problemen met seriële lijnen oplossen

## spraakpoort weergeven

toon spraak poortsleuf-nummer/poort-Gebruik deze opdracht om de poortstatus en de parameters weer te geven die op de spraak-poort van Cisco spraak-interfacekaarten (VIC) zijn geconfigureerd. Als alle Cisco IOS-opdrachten worden de standaardwaarden niet weergegeven in show in werking stelt-configureren, maar worden ze met deze opdracht wel weergegeven.

Dit is voorbeelduitvoer voor een E&M spraakpoort:

```
<#root>

router#

show voice port 1/0:1

recEive and transMit Slot is 1, Sub-unit is 0, Port is 1

Type of VoicePort is E&M
Operation State is DORMANT
Administrative State is UP

No Interface Down Failure
Description is not set
Noise Regeneration is enabled
Non Linear Processing is enabled
Music On Hold Threshold is Set to -38 dBm

In Gain is Set to 0 dB
Out Attenuation is Set to 0 dB
Echo Cancellation is enabled
Echo Cancel Coverage is set to 16 ms

Connection Mode is normal
Connection Number is not set
Initial Time Out is set to 10 s

Interdigit Time Out is set to 10 s

Call-Disconnect Time Out is set to 60 s

Region Tone is set for US


Voice card specific Info Follows:
Out Attenuation is Set to 0 dB
Echo Cancellation is enabled
Echo Cancel Coverage is set to 16 ms

Connection Mode is normal (could be trunk or plar)

Connection Number is not set
```

```
Initial Time Out is set to 10 s
Interdigit Time Out is set to 10 s
Call-Disconnect Time Out is set to 60 s
Region Tone is set for US

Voice card specific Info Follows:

Signal Type is wink-start
Operation Type is 2-wire
E&M Type is 1
Dial Type is dtmf
In Seizure is inactive
Out Seizure is inactive

Digit Duration Timing is set to 100 ms

InterDigit Duration Timing is set to 100 ms
Pulse Rate Timing is set to 10 pulses/second
InterDigit Pulse Duration Timing is set to 500 ms
Clear Wait Duration Timing is set to 400 ms
Wink Wait Duration Timing is set to 200 ms
Wink Duration Timing is set to 200 ms
Delay Start Timing is set to 300 ms
Delay Duration Timing is set to 2000 ms
Dial Pulse Min. Delay is set to 140 ms
```

## debug vpm (voice processor module)

Deze opdrachten worden gebruikt om de VPM-telefonieinterface te debuteren:

- debug vpm signaal —Deze opdracht wordt gebruikt om debug informatie voor signalering gebeurtenissen te verzamelen en kan nuttig zijn om problemen met signalering aan een PBX op te lossen.
- debug vpm spi —Deze opdracht traceert hoe de Voice Port Module Service Provider interface (SPI) met de Call Control API interfaceert. Dit debug bevel toont informatie over hoe elke netwerkaanwijzing en toepassingsverzoek wordt behandeld.
- debug vpm dsp —Deze opdracht geeft berichten weer van de DSP op de VPM naar de router en kan handig zijn als u vermoedt dat de VPM niet functioneel is. Het is een eenvoudige manier om te controleren of de VPM reageert op off-hook indicaties en om timing te evalueren voor het signaleren van berichten van de interface.
- debug vpm all —Deze EXEC opdracht maakt alle debug vpm commando's mogelijk: debug vpm spi, debug vpm signaal, en debug vpm dsp.
- debug vpm poort —Gebruik deze opdracht om de debug uitvoer naar een bepaalde poort te beperken. Deze output laat bijvoorbeeld alleen debug vpm-dspmessage zien voor poort 1/0/0:

```
debug vpm dsp
```

```
debug vpm port 1/0/0
```

Voorbeeld uitvoer voor debug vpm signalCommand

```
<#root>
maui-voip-austin#
debug vpm signal


!--- FXS port 1/0/0 goes from the "on-hook" to "off-hook" !--- state.

htsp_process_event: [1/0/0, 1.2 , 36]
fxsls_onhook_offhook htsp_setup_ind

*Mar 10 16:08:55.958: htsp_process_event:
[1/0/0, 1.3 , 8]


!--- Sends ringing alert to the called phone.

*Mar 10 16:09:02.410: htsp_process_event:
[1/0/0, 1.3 , 10] htsp_alert_notify
*Mar 10 16:09:03.378: htsp_process_event:
[1/0/0, 1.3 , 11]


!--- End of phone call, port goes "on-hook".

*Mar 10 16:09:11.966: htsp_process_event:
[1/0/0, 1.3 , 6]
*Mar 10 16:09:17.218: htsp_process_event:
[1/0/0, 1.3 , 28]
fxsls_offhook_onhook

*Mar 10 16:09:17.370: htsp_process_event:
[1/0/0, 1.3 , 41] fxsls_offhook_timer
*Mar 10 16:09:17.382: htsp_process_event:
[1/0/0, 1.2 , 7]
fxsls_onhook_release
```

Als de aan-haak en de uit-haak niet goed signaleren, controleer deze punten:

- Controleer of de bekabeling juist is.
- Controleer of zowel de router als de switch (CO of PBX) goed geaard zijn.
- Controleer of beide uiteinden van de verbinding gelijkwaardige signaleringsconfiguraties hebben. Onaangepaste configuraties kunnen onvolledige of unidirectionele signalering veroorzaken.

Zie Analoge E&M-interfacetypen en bedradingsregelingen voor meer informatie over probleemoplossing in E&M.

Voorbeeld-uitvoer voor debug vpm spiCommand

```
<#root>

maui-voip-austin#

debug vpm spi


Voice Port Module Session debugging is enabled


!--- The DSP is put into digit collection mode.

*Mar 10 16:48:55.710:

dsp_digit_collect_on:
[1/0/0]

 packet_len=20 channel_id=128
 packet_id=35 min_inter_delay=290
 max_inter_delay=3200 mim_make_time=18 max_make
_time=75 min_brake_time=18 max_brake_time=75
```

## Controleer Ontvangen en verzonden cijfers (POTS-gespreksleider)

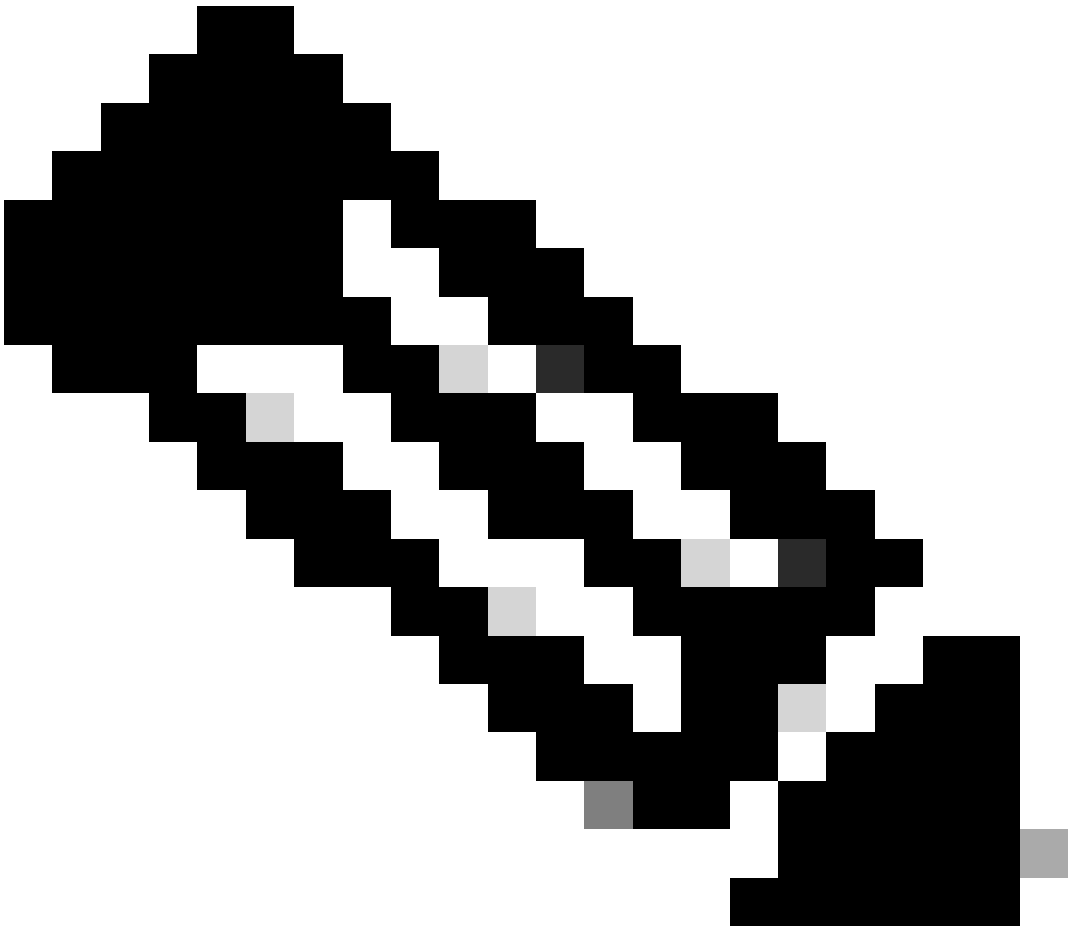
Zodra de op-haak en van-haak signalering worden geverifieerd en correct werken, verifieer de correcte cijfers worden ontvangen of op de spraak-poort verzonden (digitaal of analoog). Een dial-peer wordt niet gekoppeld of de switch (CO of PBX) kan het juiste station niet bellen als er onvolledige of onjuiste cijfers worden verzonden of ontvangen. Sommige opdrachten die kunnen worden gebruikt om de ontvangen/verzonden cijfers te verifiëren zijn:

- toon dialplan nummer-Dit bevel wordt gebruikt om te tonen welke wijzerplaatpeer wordt bereikt wanneer een bepaald telefoonnummer wordt gedraaid.
- debug vtsp sessie —Deze opdracht geeft informatie weer over hoe elk netwerkindicatie en aanvraag wordt verwerkt, signaleringsindicaties en DSP-controleberichten.
- debug vtsp dsp —Eerder dan Cisco IOS-softwarerelease 12.3, geeft deze opdracht de cijfers weer zoals deze door de spraakpoort worden ontvangen. In Cisco IOS-softwarerelease 12.3 en hoger worden de cijfers echter niet meer weergegeven door de uitvoer van de opdracht debug. De combinatie van debug hpi detail en debug hpinotification kan worden gebruikt om de inkomende cijfers te zien.
- debug vtsp all —Deze opdracht maakt deze debug Voice Telephony Service Provider (VTSP) opdrachten: debug vtsp sessie, debug vtsp fout , en debug vtsp dsp.

nummering weergeven

toon dialplan nummer <digit_string>—Deze opdracht geeft de dial-peer weer die wordt gekoppeld aan een reeks cijfers. Als meerdere dial-peers kunnen worden gekoppeld, worden ze allemaal weergegeven in de volgorde waarin ze worden gekoppeld.

---



Opmerking: U moet het #-teken aan het einde van telefoonnummers gebruiken voor dial-peers met variabele lengte om aan te sluiten op bestemmingspatronen die met T eindigen.

---

De uitvoer van deze opdracht ziet er als volgt uit:

```
<#root>
maui-voip-austin#
show dialplan number 5000
```

```
Dial string terminator: #
Macro Exp.: 5000

VoiceOverIpPeer2
        information type = voice,

        tag = 2, destination-pattern = `5000',

        answer-address = `', preference=0,
        group = 2,

Admin state is up, Operation
        state is up,

        incoming called-number = `',
        connections/maximum = 0/unlimited,
        application associated:


type = voip, session-target =
        `ipv4:192.168.10.2'

,
        technology prefix:


ip precedence = 5

, UDP checksum =
        disabled, session-protocol = cisco,
        req-qos = best-effort,
        acc-qos = best-effort,
        dtmf-relay = cisco-rtp,


fax-rate = voice,
        payload size =  20 bytes
        codec = g729r8,
        payload size =  20 bytes

,
        Expect factor = 10, Icpif = 30,
        signaling-type = cas,


VAD = enabled

, Poor QOV Trap = disabled,
        Connect Time = 25630, Charged Units = 0,
        Successful Calls = 25, Failed Calls = 0,
        Accepted Calls = 25, Refused Calls = 0,
        Last Disconnect Cause is "10  ",
        Last Disconnect Text is "normal call
        clearing.",
        Last Setup Time = 84427934.


Matched: 5000   Digits: 4
        Target: ipv4:192.168.10.2
```

debug vtsp sessie

Het debug vtsp sessie commando toont informatie over hoe de router interageert met de DSP op basis van de signaleringsindicaties uit de signaleringsstack en verzoeken van de applicatie. Dit debug bevel toont informatie over hoe elke netwerkaanwijzing en toepassingsverzoek wordt behandeld, signalerende indicaties, en DSP controleberichten.

```
<#root>

maui-voip-austin#

debug vtsp session


Voice telephony call control session debugging is on


!--- Output is suppressed.
!--- ACTION: Caller picked up handset.
!--- The DSP is allocated, jitter buffers, VAD
!--- thresholds, and signal levels are set.


*Mar 10 18:14:22.865:

dsp_set_playout

: [1/0/0 (69)]
packet_len=18 channel_id=1 packet_id=76 mode=1

initial=60 min=4 max=200 fax_nom=300

*Mar 10 18:14:22.865:

dsp_echo_canceller_control

:
[1/0/0 (69)] packet_len=10 channel_id=1 packet_id=66
flags=0x0
*Mar 10 18:14:22.865:

dsp_set_gains

: [1/0/0 (69)]
packet_len=12 channel_id=1 packet_id=91

in_gain=0 out_gain=65506

*Mar 10 18:14:22.865:

dsp_vad_enable

: [1/0/0 (69)]
packet_len=10 channel_id=1 packet_id=78

thresh=-38

act_setup_ind_ack
*Mar 10 18:14:22.869:

dsp_voice_mode

: [1/0/0 (69)]
packet_len=24 channel_id=1 packet_id=73 coding_type=1
 voice_field_size=80
```

```
VAD_flag=0 echo_length=64 comfort_noise=1


inband_detect=1

digit_relay=2


AGC_flag=0act_setup_ind_ack(): dsp_dtmf_mod
e()act_setup_ind_ack: passthru_mode = 0,
no_auto_switchover = 0dsp_dtmf_mode
(VTSP_TONE_DTMF_MODE)


!--- The DSP is put into "voice mode" and dial-tone is
!--- generated.


*Mar 10 18:14:22.873:

dsp_cp_tone_on

: [1/0/0 (69)]
packet_len=30 channel_id=1 packet_id=72 tone_id=4
n_

freq=2 freq_of_first=350 freq_of_second=440

 amp_of_first=
4000 amp_of_second=4000 direction=1 on_time_first=65535
off_time_first=0 on_time
_second=65535 off_time_second=0
```

Als wordt vastgesteld dat de cijfers niet goed worden verzonden of ontvangen, kan het mogelijk nodig zijn om een digit-grabber (testgereedschap) of T1-tester te gebruiken om te controleren of de cijfers met de juiste frequentie en timing-interval worden verzonden. Als deze "onjuist" voor de switch (CO of PBX) worden verzonden, kunnen bepaalde waarden op de router of switch (CO of PBX) mogelijk moeten worden aangepast zodat ze overeenkomen en onderling kunnen werken. Dit zijn gewoonlijk cijfers duur en de waarden van de inter-cijferduur. Een ander item dat moet worden onderzocht als de cijfers correct lijken te worden verzonden, zijn alle nummervertaaltabellen in de switch (CO of PBX) die cijfers kunnen toevoegen of verwijderen.

## Controleer end-to-end VoIP-signalering (VOIP-gesprekssignalering)

Nadat u hebt geverifieerd dat spraak-poorts signalering correct werkt en de juiste cijfers worden ontvangen, gaat u naar de VoIP-gesprekscontrole voor probleemoplossing en debuggen. Deze factoren verklaren waarom het debuggen van gespreksbeheer een complexe taak kan worden:

- Cisco VoIP-gateways gebruiken H.323-signalering om gesprekken te voltooien. H.323 bestaat uit drie lagen van gespreksonderhandeling en gespreksinstelling: H.225, H.245 en H.323. Deze protocollen maken gebruik van een combinatie van TCP en UDP om een aanroep in te stellen en in te stellen.

- End-to-end VoIP-debugging laat een aantal Cisco IOS-statustools zien. Problemen met een state-machine kunnen ervoor zorgen dat een oproep mislukt.
- End-to-end VoIP-debugging kan zeer ruim zijn en veel debug-uitvoer opleveren.

## debug voip capi inout

De primaire opdracht om end-to-end VoIP-oproepen te debuggen is debug voip capi inout . De output van een vraag zuivert wordt getoond in deze output.

```
<#root>

!--- Action: A VoIP call is originated through the
!--- Telephony SPI (pots leg) to extension 5000.
!--- Some output is omitted.


maui-voip-austin#

debug voip ccapi inout

voip ccAPI function enter/exit debugging is on


!--- Call leg identification, source peer: Call
!--- originated from dial-peer 1 pots
!--- (extension 4000).


*Mar 15 22:07:11.959: cc_api_call_setup_ind
(vdbPtr=0x81B09EFC,

callInfo={called=,
calling=4000, fdest=0 peer_tag=1

}, callID=0x81B628F0)

!--- CCAPI invokes the session application.



*Mar 15 22:07:11.963: cc_process_call_setup_ind
(event=0x81B67E44) handed call to app "SESSION"

*Mar 15 22:07:11.963: sess_appl:
ev(23=CC_EV_CALL_SETUP_IND), cid(88), disp(0)


!--- Allocate call leg identifiers "callid = 0x59"


*Mar 15 22:07:11.963: ccCallSetContext
(

callID=0x58

, context=0x81BAF154)
*Mar 15 22:07:11.963: ccCallSetupAck
```

```
(
callID=0x58
)

!--- Instruct VTSP to generate dialtone
.

*Mar 15 22:07:11.963: ccGenerateTone
(callID=0x58
 tone=8)

!--- VTSP passes digits to CCAPI.

*Mar 15 22:07:20.275:cc_api_call_digit_begin
(vdbPtr=0x81B09EFC,callID=0x58,digit=5, flags=0x1, timestamp=0xC2E63BB7, expiration=0x0)
*Mar 15 22:07:20.279: sess_appl:
ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(88), disp(0)
*Mar 15 22:07:20.279: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0)
*Mar 15 22:07:20.279: ssaIgnore cid(88),
st(0),oldst(0), ev(10)
*Mar 15 22:07:20.327: cc_api_call_digit
(vdbPtr=0x81B09EFC, callID=0x58, digit=5
, duration=100)
*Mar 15 22:07:20.327: sess_appl:
ev(9=CC_EV_CALL_DIGIT), cid(88), disp(0)
*Mar 15 22:07:20.327: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDes
t(0)
*Mar 15 22:07:21.975:cc_api_call_digit_begin
(vdbPtr=0x81B09EFC,callID=0x58,digit=0,
flags=0x1, timestamp=0xC2E63BB7, expiration=0x0)
*Mar 15 22:07:21.979: sess_appl:
ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(88), disp(0)
*Mar 15 22:07:21.979: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDes
t(0)
*Mar 15 22:07:21.979: ssaIgnore cid(88),
st(0),oldst(0), ev(10)
*Mar 15 22:07:22.075: cc_api_call_digit
(vdbPtr=0x81B09EFC, callID=0x58, digit=0
, duration=150)
*Mar 15 22:07:22.079: sess_appl:
ev(9=CC_EV_CALL_DIGIT), cid(88), disp(0)
*Mar 15 22:07:22.079: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0)
*Mar 15 22:07:23.235: cc_api_call_digit_begin
(vdbPtr=0x81B09EFC, callID=0x58, dgit=0,
flags=0x1, timestamp=0xC2E63BB7, expiration=0x0)
*Mar 15 22:07:23.239: sess_appl:
ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(88), disp(0)
*Mar 15 22:07:23.239: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0)
```

```
*Mar 15 22:07:23.239: ssaIgnore cid(88),
st(0),oldst(0), ev(10)

*Mar 15 22:07:23.335: cc_api_call_digit
(vdbPtr=0x81B09EFC, callID=0x58, digit=0

, duration=150)
*Mar 15 22:07:23.339: sess_appl:
ev(9=CC_EV_CALL_DIGIT), cid(88), disp(0)
*Mar 15 22:07:23.339: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDes
t(0)
*Mar 15 22:07:25.147: cc_api_call_digit_begin
(vdbPtr=0x81B09EFC, callID=0x58, d
igit=0, flags=0x1, timestamp=0xC2E63BB7,
expiration=0x0)
*Mar 15 22:07:25.147: sess_appl:
ev(10=CC_EV_CALL_DIGIT_BEGIN), cid(88), disp(0)
*Mar 15 22:07:25.147: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0)
*Mar 15 22:07:25.147: ssaIgnore cid(88),
st(0),oldst(0), ev(10)

*Mar 15 22:07:25.255: cc_api_call_digit
(vdbPtr=0x81B09EFC, callID=0x58, digit=0

, duration=160)
*Mar 15 22:07:25.259: sess_appl:
ev(9=CC_EV_CALL_DIGIT), cid(88), disp(0)
*Mar 15 22:07:25.259: ssaTraceSct:
cid(88)st(0)oldst(0)cfid(-1)csize(0)in(1)fDest(0)


!--- Matched dial-peer 2 voip. Destination number !--- 5000



*Mar 15 22:07:25.259: ssaSetupPeer cid(88)
peer list:tag(2) called number(5000)


*Mar 15 22:07:25.259: ssaSetupPeer cid(88),

destPat(5000)

, matched(4), prefix(),
peer(81C04A10)


!--- Continue to call an interface and start the !--- next call leg.



*Mar 15 22:07:25.259: ccCallProceeding
(callID=0x58

, prog_ind=0x0)

*Mar 15 22:07:25.259: ccCallSetupRequest
(Inbound call = 0x58, outbound peer =2,
dest=, params=0x81BAF168 mode=0,
*callID=0x81B6DE58)
*Mar 15 22:07:25.259: callingNumber=4000,
calledNumber=5000
```

```
, redirectNumber=


!--- VoIP call setup.



*Mar 15 22:07:25.263: ccIFCallSetupRequest:

(vdbPtr=0x81A75558, dest=,

callParams={called=5000, calling=4000,
fdest=0, voice_peer_tag=2}

, mode=0x0)

*Mar 15 22:07:25.263: ccCallSetContext
(callID=0x59

, context=0x81BAF3E4)
*Mar 15 22:07:25.375: ccCallAlert
(callID=0x58, prog_ind=0x8, sig_ind=0x1)


!--- POTS and VoIP call legs are tied together.



*Mar 15 22:07:25.375: ccConferenceCreate
(confID=0x81B6DEA0, callID1=0x58, callI
D2=0x59, tag=0x0)

*Mar 15 22:07:25.375: cc_api_bridge_done


(confID=0x1E, srcIF=0x81B09EFC,

srcCall

ID=0x58, dstCallID=0x59

, disposition=0,
tag=0x0)


!--- Exchange capability bitmasks with remote
!--- the VoIP gateway
!--- (Codec, VAD, VoIP or FAX, FAX-rate, and so forth).



*Mar 15 22:07:26.127: cc_api_caps_ind

(dstVdbPtr=0x81B09EFC,

dstCallId=0x58, src
CallId=0x59,caps={codec=0x4, fax_rate=0x2,
vad=0x2, modem=0x1 codec_bytes=20,
signal_type=0})



!--- Both gateways agree on capabilities.


*Mar 15 22:07:26.127: cc_api_caps_ack
```

```
(dstVdbPtr=0x81B09EFC,

dstCallId=0x58, src

CallId=0x59, caps={codec=0x4, fax_rate=0x2,
vad=0x2, modem=0x1 codec_bytes=20,
signal_type=0})

*Mar 15 22:07:26.139: cc_api_caps_ack


(dstVdbPtr=0x81A75558,

dstCallId=0x59

, src

CallId=0x58, caps={codec=0x4, fax_rate=0x2,
vad=0x2, modem=0x1 codec_bytes=20,
signal_type=0})

*Mar 15 22:07:35.259: cc_api_call_digit
(vdbPtr=0x81B09EFC, callID=0x58, digit=T
, duration=0)
*Mar 15 22:07:35.259: sess_appl:
ev(9=CC_EV_CALL_DIGIT), cid(88), disp(0)
*Mar 15 22:07:35.259: ssaTraceSct:
cid(88)st(4)oldst(3)cfid(30)csize(0)in(1)
fDest(0)-cid2(89)st2(4)oldst2(1)

*Mar 15 22:07:35.399: cc_api_call_connected

(vdbPtr=0x81A75558, callID=0x59)
*Mar 15 22:07:35.399: sess_appl:
ev(8=CC_EV_CALL_CONNECTED), cid(89), disp(0)
*Mar 15 22:07:35.399: ssaTraceSct:
cid(89)st(4)oldst(1)cfid(30)csize(0)in(0)
fDest(0)-cid2(88)st2(4)oldst2(4)


!--- VoIP call is connected.



*Mar 15 22:07:35.399: ccCallConnect


(callID=0x58)


!--- VoIP call is disconnected. Cause = 0x10



*Mar 15 23:29:39.530: ccCallDisconnect
(callID=0x5B, cause=0x10 tag=0x0)
```

Als de oproep mislukt en de oorzaak lijkt te liggen in het VoIP-gedeelte van de gespreksinstallatie, kunt u mogelijk het H.225 of H.245 TCP-gedeelte van de gespreksinstallatie bekijken, in tegenstelling tot alleen het UDP-gedeelte van de H.323-installatie. De opdrachten die kunnen worden gebruikt om de instellingen van de H.225- of H.245-oproep te debuggen zijn:

- debug ip tcp transacties en debug ip tcp pakket-deze opdrachten onderzoeken het TCP gedeelte van de H.225 en H.245 onderhandeling. Ze geven de IP adressen, TCP poorten en statussen van de TCP verbindingen terug.
- debug cch323 h225 —Deze opdracht onderzoekt het H.225-gedeelte van de gespreksonderhandeling en traceert de statustransitie van de H.225-statesmachine op basis van de verwerkte gebeurtenis. Zie dit als Layer 1 deel van de driedelige H.323 gespreksinstallatie.
- debug cch323 h245 —Deze opdracht onderzoekt het H.245-gedeelte van de gespreksonderhandeling en traceert de staatsovergang van de H.245-toestandsmachine op basis van de verwerkte gebeurtenissen. Zie dit als Layer 2 deel van de drieluik H.323 gespreksinstallatie.

## VoIP Quality of Service (QoS)-problemen begrijpen

Wanneer VoIP-oproepen op de juiste manier tot stand worden gebracht, moet als volgende stap worden geverifieerd dat de spraakkwaliteit goed is. Hoewel het oplossen van QoS-problemen niet in dit document wordt behandeld, moeten deze richtlijnen worden overwogen om een goede spraakkwaliteit te bereiken:

- Begrijp hoeveel bandbreedte een VoIP vraag met elke codec verbruikt. Dit omvat Layer 2- en IP/UDP/RTP-headers. Raadpleeg Bandbreedteconsumptie wijzigen voor spraakoproepen voor meer informatie.
- Begrijp de kenmerken van het IP netwerk waar de vraag over reist. De bandbreedte van een Frame Relay-netwerk bij CIR is bijvoorbeeld veel anders dan die boven-CIR (of burst), waar pakketten kunnen worden gedropt of in de Frame Relay-cloud worden opgeslagen. Zorg ervoor dat vertraging en jitter zoveel mogelijk worden beheerst en geëlimineerd. De eenrichtingsvertraging van de transmissie mag niet langer zijn dan 150 ms (volgens een G.114-aanbeveling).
- Gebruik een wachtrijtechniek waarmee VoIP-verkeer kan worden geïdentificeerd en prioriteit kan krijgen.
- Wanneer u VoIP via snelle links overbrengt, gebruikt u Layer 2-pakketfragmentatietechnieken, zoals MLPPP met Link Fragmentation and Interleaving (LFI) op point-to-point links of FRF.12 op Frame Relay-links. De fragmentatie van grotere gegevenspakketten staat minder jitter en vertraging in het overbrengen van VoIP verkeer toe omdat de pakketten VoIP op de verbinding kunnen worden doorschoven.
- Probeer een andere codec te gebruiken en probeer de oproep met VAD ingeschakeld en uitgeschakeld om de kwestie mogelijk te beperken tot de DSP, in tegenstelling tot het IP-netwerk.

Met VoIP zijn de belangrijkste dingen om te zoeken wanneer problemen met QoS oplossen verloren pakketten en netwerk knelpunten die vertraging en jitter kunnen veroorzaken.

Zoek naar:

- interface-druppels
- bufferdalingen
- interfacestremming
- koppelingscongestie

Elke interface in het pad van de VoIP-oproep moet worden onderzocht. Ook, elimineer druppels en congestie. Ook moet de vertraging van retourvluchten zoveel mogelijk worden beperkt. Pings tussen de eindpunten van VoIP geven een aanwijzing van de ronde reisvertraging van een verbinding. De vertraging van de retourvlucht mag niet meer dan 300 ms bedragen. Indien de vertraging deze waarde moet overschrijden, moet ernaar worden gestreefd dat deze vertraging constant blijft, zodat er geen trillingen of wisselende vertragingen ontstaan.

Er moet ook verificatie worden uitgevoerd om ervoor te zorgen dat het Cisco IOS-wachtmechanisme de VoIP-pakketten in de juiste wachtrijen plaatst. Cisco IOS-opdrachten, zoals interface in wachtrij tonen of prioriteit tonen, kunnen helpen bij de verificatie van wachtrij.

# Details van Cause Codes en Debug VoIP waarden

Gebruik deze tabellen wanneer u debugs leest en de bijbehorende waarden binnen de debugs.

## Q.931 Oorzaken van gespreksonderbreking (cause_codes van debug voip capi inout)

| Waarde van gespreksonderbreking (in hex) | Betekenis en aantal (in decimalen) |
|---|---|
| C_OORZAAK_UANUM = 0x1 | niet-toegewezen nummer. (1) |
| CC_OORZAAK_NO_ROUTE = 0x3 | geen route naar bestemming. (3) |
| C_OORZAAK_NORM = 0x10 | normale gespreksverwijdering. (16) |
| C_OORZAAK_BEZIG = 0x11 | gebruiker bezig. (17) |
| C_OORZAAK_NORS = 0x12 | geen gebruikersreactie. (18) |
| C_OORZAAK_NOAN = 0x13 | geen gebruikersantwoord. (19) |
| C_OORZAAK_AFSTOTING = 0x15 | afwijzing van de uitnodiging. (21) |
| CC_OORZAAK_ONGELDIG_NUMMER = 0x1C | ongeldig nummer. (28) |
| C_OORZAAK_UNSP = 0x1F | normaal, niet gespecificeerd. (31) |
| CC_OORZAAK_NO_CIRCUIT = 0x22 | geen circuit. (34) |
| CC_OORZAAK_NO_REQ_CIRCUIT = 0x2C | geen vereist circuit. (44) |
| C_OORZAAK_NO_RESOURCE = 0x2F | geen middelen. (47)[1] |
| CC_OORZAAK_NOSV = 0x3F | de service of optie is niet beschikbaar of niet gespecificeerd. (63) |

[1]Dit probleem kan ontstaan door een codec-fout in de H323-instelling. De eerste stap van de probleemoplossing is dus om de VoIP-dial-peers te coderen om de juiste codec te gebruiken.

## Codec-onderhandelingswaarden (van debug voip capi inout)

Zie Codecs begrijpen, voor meer informatie over Codecs: Complexity, Hardware Support, MOS en Onderhandeling.

| Onderhandelingswaarde | Betekenis |
|---|---|
| codec=0x00000001 | G711 ULAW 64K PCM |
| codec=0x00000002 | G711 ALAW 64K PCM |
| codec=0x00000004 | 729 |
| codec=0x00000004 | G729IETF |
| codec=0x00000008 | G729a |
| codec=0x00000010 | G726r16 router |
| codec=0x00000020 | G726r2 |
| codec=0x00000040 | G726r32 |
| codec=0x00000080 | 728 |
| codec=0x00000100 | G723r63 router |
| codec=0x00000200 | G723r5 |
| codec=0x00000400 | GSMFR |
| codec=0x00000800 | G729b |
| codec=0x00001000 | G729ab |
| codec=0x00002000 | G723ar63 |
| codec=0x00004000 | G723ar53 |
| codec=0x00008000 | CLEAR_CHANNEL |

## Tone-typen

| Tone-typen | Betekenis |
|---|---|
| CC_TONE_RINGBACK 0x1 | Ringtoon |
| CC_TONE_FAX 0x2 | Faxtoon |
| CC_TONE_BUSY 0x4 | Bezettoon |
| CC_TONE_DIALTONE 0x8 | Kiestoon |
| CC_TONE_OS 0x10 | Tone buiten bedrijf |
| CC_TONE_ADDR_ACK_0x20 | Adresbevestigingstoon |
| CC_TONE_DISCONNECT 0x40 | Toon loskoppelen |
| CC_TONE_OFF_HOOK_OPMERKING 0x80 | Toon die aangeeft dat de telefoon niet was aangesloten op de haak |

| | |
|---|---|
| CC_TONE_OFF_HOOK_ALERT 0x100 | Een urgentere versie van CC_TONE_OFF_HOOK_notice |
| CC_TONE_AANGEPAST 0x200 | Aangepaste tint - gebruikt wanneer u een aangepaste tint opgeeft |
| CC_TONE_NULL 0x0 | Ongeldige tint |

Waarden voor fax- en VAD-mogelijkheden

| Waarden | Betekenis |
|---|---|
| CC_CAP_FAX_GEEN 0x1 | Fax schakelt uit of niet beschikbaar |
| CC_CAP_FAX_SPRAAK 0x2 | Spraakoproep |
| CC_CAP_FAX_144 0x4 | 14.400 baud |
| CC_CAP_FAX_96 0x8 | 9.600 baud |
| CC_CAP_FAX_72 0x10 | 7 200 baud |
| CC_CAP_FAX_48 0x20 | 4.800 baud |
| CC_CAP_FAX_24 0x40 | 2.400 baud |
| CC_CAP_VAD_OFF 0x1 | VAD uitgeschakeld |
| CC_CAP_VAD_ON 0x2. | VAD ingeschakeld |

## Gerelateerde informatie

- T1 Layer 1-probleemoplossing
- T1-probleemoplossing
- Problemen met seriële lijnen oplossen
- Probleemoplossing voor Cisco IP-telefonie
- Cisco Technical Support en downloads

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.